

# Towards Open-World Gesture Recognition

Junxiao Shen\*  
Meta Reality Labs Research  
University of Bristol

Ran Tan  
Meta Reality Labs Research

Matthias De Lange  
Meta Reality Labs Research

Naveen Suda  
Meta Reality Labs Research  
Amy Karlson  
Meta Reality Labs Research

Xuhai Xu  
Meta Reality Labs Research

Maciej Lazarewicz  
Meta Reality Labs Research  
Evan Strasnick  
Meta Reality Labs Research

Enmin Zhou  
Meta Reality Labs Research

Per Ola Kristensson  
University of Cambridge

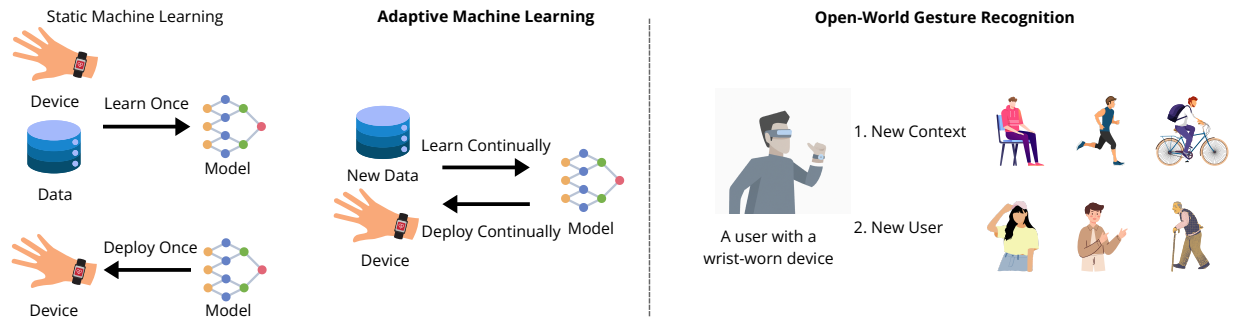


Figure 1: **Static Machine Learning:** The model undergoes a single training session using existing data and is subsequently deployed to the device. **Adaptive Machine Learning:** Recognizing the dynamic nature of real-world scenarios, this approach continuously updates and refines the model as new data become available. Consequently, updated models are regularly deployed to the device. **Open-World Gesture Recognition:** In the context of wrist-worn devices utilized for gesture recognition, there are two distinct cases where new data will appear. In these cases, the gesture recognition model must adapt and learn from this new information.

## ABSTRACT

Providing users with accurate gestural interfaces, such as gesture recognition based on wrist-worn devices, is a key challenge in mixed reality. However, static machine learning processes in gesture recognition assume that training and test data come from the same underlying distribution. Unfortunately, in real-world applications involving gesture recognition, such as gesture recognition based on wrist-worn devices, the data distribution may change over time. We formulate this problem of adapting recognition models to new tasks, where new data patterns emerge, as *open-world gesture recognition* (OWGR). We propose the use of continual learning to enable machine learning models to be adaptive to new tasks without degrading performance on previously learned tasks. However, the process of exploring parameters for questions around when, and how, to train and deploy recognition models requires resource-intensive user studies may be impractical. To address this challenge, we propose a design engineering approach that enables offline analysis on a collected large-scale dataset by systematically examining various parameters and comparing different continual learning methods. Finally, we provide design guidelines to enhance the development of an open-world wrist-worn gesture recognition process.

**Index Terms:** Computing methodologies—Machine learning—Learning paradigms—Lifelong machine learning; Human-centered computing—Human computer interaction (HCI)—HCI design and evaluation methods—User models

\*e-mail: junxiao.shen@bristol.ac.uk

## 1 INTRODUCTION

Wrist-worn gesture recognition using inertial measurement unit (IMU) signals offers a convenient, always-on interface for various applications in mixed reality [21, 42, 46, 54]. Currently, most hand gesture recognition algorithms are optimized for *closed-world* settings, where training and test data come from the same underlying distribution [53]. However, in real-world applications [3, 43], the assumption that training and test data belong to the same underlying distribution no longer holds as: (1) new gesture data continuously arrives with changing characteristics; (2) gesture data may change over time; and (3) entirely new data patterns can emerge. We here call this problem *open-world gesture recognition* (OWGR). A static machine learning process, in which a recognition model is trained once and subsequently deployed, cannot effectively tackle this problem. What we seek is an *adaptive* machine learning process that can continuously train and deploy the model on newly *emerging* data, as illustrated in Figure 1.

Various approaches can be used for adaptive machine learning. Simply retraining a recognition model with the entire joint dataset, past data and new data from new tasks, is sometimes infeasible due to limited computational power on embedded devices. Storing the entire past dataset is also challenging due to the limited memory of the device and privacy concerns. On the other hand, if a model trained from past data is naively fine-tuned on a new task, that model will dramatically decrease the recognition performance on the old tasks [47]. This is called catastrophic forgetting. Continual learning, sometimes called lifelong learning, methods are specifically designed to alleviate catastrophic forgetting by balancing the trade-off between plasticity (transferring knowledge from an old task to a new task) and stability (catastrophic forgetting).

We identify two real-world cases that fall under OWGR, *new context* and *new user*, as shown in Figure 1. In all two cases, we

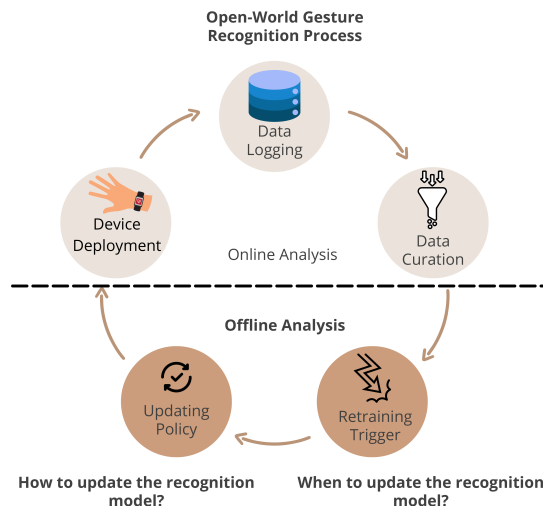


Figure 2: The *open-world gesture recognition* process is structured around five stages. Out of these, two stages including *retraining trigger* and *updating policy* introduce essential design considerations, all underpinned by an engineering approach that facilitates offline analysis. In contrast, the other three stages including *device deployment*, *data logging* and *data curation* require online analysis.

wish to preserve performance on previously learned tasks, to avoid catastrophic forgetting, while at the same time learning the new task. Here, a *new task* may refer to a new context, or a new user. In Section 3, we present a key contribution in this paper—a detailed formulation of the *open-world gesture recognition* problem and its associated real-world scenarios, in terms of these two cases: (1) new context; and (2) new user.

In a static machine learning process, we only need to log the data from the device, optimize the recognition model on the logged data to best fit the data once, and then deploy a trained model onto the device. However, in an adaptive machine learning process for OWGR, we additionally introduce additional stages as illustrated in Figure 2. In particular, this OWGR process includes two additional stages that require careful design considerations: (1) *when* to update the model; and (2) *how* to update the model. When refining a wrist-worn *open-world gesture recognition* system, developers face challenges due to the complexity of user studies and numerous parameters. For instance, evaluating *new context* requires recruiting participants to perform gestures in various real-world contexts, not just in a lab. Additionally, optimizing the OWGR process often needs multiple evaluations, making it difficult to execute and validate.

We therefore follow a prior approach [25], which proposes using a design engineering approach to assess a process that is challenging to evaluate via user studies. We have adopted this design engineering approach by translating it into a strategy that outlines a functional design of *open-world gesture recognition* and examines its conceptual design in which the key function of the process (such as *Updating Policy*) is translated into six potential function carriers (five different implementations of continual learning methods and one finetuning baseline). We then distinguish between controllable and uncontrollable parameters of these functions and assess the process’s effectiveness through quantitative envelope analysis. Moreover, given the lack of prior research that focuses specifically on OWGR, it is crucial to first apply the most advanced and generalizable continual learning methods to address this issue. Only after extensive exploration and evaluation of OWGR using other more generalizable and well-established continual learning methods are we in a position to best understand how to propose new algorithms

to further enhance the system. Therefore, proposing a new algorithm is beyond the scope of this paper.

Such a design engineering approach allows us to assess controllable and uncontrollable parameters offline, thus sidestepping the need for on-ground deployments and tests with real users in the first instance. To this end, we first collect a large-scale dataset using the IMU sensor from wrist-worn devices. To capture diversity in the aforementioned two real-world cases, the dataset contains 50 users performing four different gestures in 25 different contexts. We then construct a surrogate data model and a surrogate task model, identify their controllable and uncontrollable parameters, and subsequently perform an envelope analysis by varying these parameters, which allows us to assess emergent qualities of the system as a whole via simulation. This approach is then able to illustrate that using a continual learning method for *open-world gesture recognition* significantly improves gesture recognition accuracy and reduces catastrophic forgetting compared to a finetuning method. This forms our second contribution to the paper.

Designers using engineering methodologies to refine the OWGR process face limitations in data collection and computational resources. Envelope analysis, being computationally intensive, demands extensive simulations and parameter sweeps. Our study required over 40,000 GPU hours for dataset collection and offline evaluation of continual learning methods, highlighting the challenge for resource-constrained designers. To aid these designers, we provide essential design guidelines derived from our methodology. Our diverse dataset, covering various users and contexts, ensures the generalizability of these guidelines. Therefore, our third contribution is offering distilled guidelines for developing OWGR processes based on our engineering approach using envelope analysis.

In summary, this paper makes three contributions: (1) a detailed formulation of the *open-world gesture recognition* (OWGR) problem; (2) a design engineering approach for allowing *open-world gesture recognition* to be tractable in practice; and (3) design guidelines for developing OWGR processes.

## 2 RELATED WORK

### 2.1 Wrist-Worn Devices with Mixed Reality

Gestural interaction offers an intuitive and natural method for interacting with mixed reality [67]. More commonly, the front-facing cameras of Head Mounted Displays (HMDs) track the hands, enabling hand gestures for text entry, virtual object manipulation, and menu interaction, such as dragging virtual items or making selections [4]. However, a fundamental drawback of using front-facing cameras is the necessity for the hands to remain within the camera’s view. This means users cannot perform gestures with their hands under the table or in their pockets. Once hands leave the camera’s tracking region, gestural interaction becomes impossible.

To overcome this limitation, wrist-worn devices have been employed for gestural interaction with mixed reality devices [2, 13]. Gesture recognition is conducted using these wrist-worn devices, eliminating the need for HMD cameras to detect hands and gestures. Using wrist-worn devices facilitates gestural interactions even when hands are under the table or in a pocket, offering a seamless user experience. Moreover, wrist-worn devices can recognize gestures in a broader variety of ways compared to front-facing cameras, which primarily rely on computer vision techniques. There has been extensive research exploring hand gesture recognition using a wide range of wrist-worn sensing modalities, such as RGB cameras [18, 64, 65], infrared (IR) ranging [35], inertial measurement unit (IMU) [17, 23], acoustics [27, 38], electromyography (EMG) [6, 36, 47], electrical impedance tomography [71], pressure [12, 22], radar [29], stretch sensors [55], magnetic sensors [9, 39], and bio-capacitive effects [57]. IMU sensors are widely used for gesture recognition owing to their low cost and low power characteristics [66, 73].

## 2.2 Gesture Recognition

As for gesture recognition techniques, early trajectory-based gesture recognition methods (e.g., dynamic time warping (DTW) [30] and hidden Markov models (HMM) [37]) can recognize simple gesture trajectories such as drawing a shape [37]). However, these methods do not work well for more complex and fine-grained gestures such as pinching or making a fist. Recently, researchers and practitioners mainly use data-driven approaches by collecting a labeled gesture dataset. They then either train traditional models such as SVM, trees, (e.g., [16, 19]) or deep learning models when the dataset is large enough (e.g., [18, 68]). However, most prior work focuses on recognizing a pre-defined gesture set. There are very few works addressing *open-world gesture recognition*. Xu *et al.* proposed a few-shot learning framework for gesture authoring [66]. Shen *et al.* proposed a deep-learning model that can learn a new gesture with a synthetic dataset that is generated from a few data samples with a deep generative model [49–51]. Wang *et al.* [62] identified a continual learning application for lifelong egocentric gesture recognition such that a VR system allows users to customize gestures incrementally. While these approaches are useful for models to learn new gestures, they are not appropriate for the two cases we identified, *new context* and *new user*. Moreover, Xu *et al.* [66] and Shen *et al.* [51] also fail to address the catastrophic forgetting problem when old gesture data are revisited.

## 2.3 Design Engineering

Design engineering is a holistic method to create products and systems. This approach transitions from the initial stage of problem identification to tackling design-related aspects during the life cycle of a product or system, including production, upkeep, and eventual decommissioning. Validating certain systems, such as a context-aware sentence retrieval system for AAC users, is extremely challenging [25] as such systems require tailored setups and extended use before benefits emerge. Asking an AAC user to switch devices, potentially waiting months for improved communication, raises both logistical and ethical concerns. Prior work [25] has used a design engineering approach to tackle this problem. This approach entails defining controllable and uncontrollable parameters for the design and examining the potential effectiveness of such systems through quantitative envelope analysis by varying the parameters through simulation [25]. Such a design engineering method reveals insights about the feasibility of such systems without the immediate need to develop, introduce, and observe systems for extended time periods. In addition, it allows examination of emergent qualities of systems that would be very difficult to study in empirical user studies, such as mechanisms explaining users strategies [26]. Similarly, Shen *et al.* [52] also used a design engineering approach to evaluate a multi-turn dialogue system for AAC using context-aware sentence generation by bag-of-keywords. The design engineering approach we propose in this paper shares a similar philosophy with this prior work.

## 2.4 Continual Learning

In a continual learning setup, the learning system is exposed to a sequence of tasks over time, each represented by a specific data distribution. The tasks may be related or completely different. The model is expected to learn new tasks while maintaining performance on previous ones, typically without access to data from earlier tasks due to memory constraints or privacy issues. When a model, previously trained on older tasks, is simply fine-tuned on new tasks, its accuracy on the old tasks rapidly deteriorates. It loses its proficiency in tasks it had previously mastered, effectively replacing the old knowledge with the new. This phenomenon is referred to as catastrophic forgetting. This problem occurs because conventional neural networks update their weights primarily based on the most recent data during training, a process that can result in the erasure

of previously acquired patterns and knowledge. More specifically, gradient-based optimization algorithms prioritize minimizing the loss of the current training task, often disregarding previous task parameter settings [15].

To address this challenge, continual learning methods aim to leverage previously seen information during training to improve performance on new tasks. The objective is to overcome the forgetting of learned tasks (stability) and utilize prior knowledge to achieve better performance and faster convergence on new tasks (plasticity) [11]. There are three prominent families of methods for continual learning: replay methods, regularization-based methods, and parameter isolation methods [11]. Replay methods [5, 8, 20, 40, 41] use rehearsal to mitigate catastrophic forgetting, storing examples from old tasks in memory to be replayed throughout incremental task training. However, these methods are less memory efficient. Regularization-based methods introduce a regularization term in the loss function of the model [1, 32, 44, 70]. Parameter isolation methods either dedicate different subsets of the model parameters to each task to prevent any possible forgetting or expand the size of the model to acquire new knowledge from new tasks [33, 45, 69]. Neither regularization-based nor parameter isolation methods store past data in memory.

There are also other different learning paradigms, such as multi-task learning [72], transfer learning [63], meta learning [58] and online learning [48]. However, while most of these methods focus on plasticity, they usually fail to address catastrophic forgetting, leading to a dramatic decrease in the performance with old tasks [11]. Being able to avoid catastrophic forgetting is critical in OWGR as it is not solely about quickly learning new data - a challenge that can be addressed by few-shot learning [56] or domain adaptation [34] - but more importantly, the core challenge of OWGR is to retain old data. More specifically, domain adaptation focuses on adapting to new domains without necessarily retaining performance on the original domain, which does not directly address catastrophic forgetting. Few-shot learning deals with learning new tasks or adapting to new data quickly from few examples, but it does not inherently address learning multiple tasks sequentially without forgetting. These two methods are specifically designed to optimize plasticity (transferring knowledge from an old task to a new task), without considering stability (catastrophic forgetting). Therefore, these learning paradigms are not well-suited to the OWGR problem. Hence, we do not directly compare continual learning methods with other learning paradigms in this paper.

## 3 OPEN-WORLD GESTURE RECOGNITION (OWGR)

### 3.1 Real-World Cases

1. *New context* is where each task represents different environmental/activity contexts, such as *standing*, *walking*, *riding in a car*, *pushing stroller/cart*, *laying down* etc. Therefore, we desire an OWGR system that is initialized on training data only from *standing* contexts to be able to incrementally learn on new contexts such as *walking* and *riding in a car*, while also preserving the knowledge from the previous *standing* contexts. These contexts can be further split into more detailed contexts. For example, *standing* includes *standing with hand up to the chest level* and *standing with hand hanging*, and *walking* includes *walking with hand up to the chest level* and *walking with hand hanging*. Our dataset includes this finer granularity of activity context. In practice, we need a *task descriptor* to inform the model of each task in the *new context*. In the real-world deployment, the description can be determined by the contextual information provided by other devices, such as location or an HMD.

2. *New User* is where each new task is a separate user of the device. A shared wrist-worn device should quickly adapt to new users, as we assume the behaviors of performing gestures from different users are different. Moreover, we also desire the device to not forget the old users as this shareable device may be switched back to old users. The task descriptor is reflected by the user’s identity.

### 3.2 Problem Formulation

We formulate the OWGR problem more formally in this section. In a *task-incremental setting for open-world gesture recognition*, the tasks (new context/gesture/user) arrive sequentially and the learner optimizes until convergence within each task [14, 41, 45]. This setting has the following assumptions:

1. Each new context/gesture/user would have a batch of data points. Therefore, the gesture data arrives sequentially in batches, with each batch corresponding to one task. The continual learning takes one batch at a time, while we can still perform offline learning *within* each task.
2. The recognition model will have a multi-head configuration, as each task needs a separate output layer. Therefore, a task descriptor is also fed into the recognition model since the algorithm needs to know which head to use for that specific task. In practice, this task descriptor would be generated through various sensing modalities (e.g., activity recognition and location information [60]). For example, a classifier using egocentric video from a head-mounted display would estimate an activity context. We discuss the feasibility of this method in Section 6.2.1 and Section 6.5. Despite a multi-head configuration may not be necessary for the *new user* case, the multi-head approach simplifies the comparative analysis of different continual learning methods — allowing for consistent evaluation with minimal adjustments.

Traditional gesture recognition assumes testing and training data share similar characteristics, that is, they are independent and identically distributed (i.i.d). For this problem setting, we represent the training set as  $D_{train} = (x_i, y_i)_{i=1}^n$ , where  $n$  denotes the number of training data available, contains a feature vector  $x_i \in X$ , and a target vector  $y_i \in Y$ . Each data pair in the training set  $(x_i, y_i)$  is sampled from an i.i.d probability distribution, which corresponds to a single task. The goal is to minimize the empirical risk of all data in the task by optimizing the parameters  $\theta$  of the gesture recognition model [59]:

$$\frac{1}{|D_{train}|} \sum_{(x_i, y_i) \in D_{train}} \mathcal{L}(f(x_i; \theta), y_i), \quad (1)$$

where the loss function  $\mathcal{L}$  penalizes prediction errors, and  $f$  denotes the trained gesture recognition model.

However, the i.i.d assumption no longer holds in OWGR, where data arrives in an online fashion with changing characteristics. A learner will observe the continuum of data as a triplet  $(x_i, y_i, t_i)$ .  $t_i \in T$  is a task descriptor identifying the task associated with the pair  $(x_i, y_i)$ . The pair is sampled from a probability distribution  $P_{t_i}$  corresponding to the task descriptor  $t_i$ . In each task  $t_i$ , the gesture data pair  $(x_i, y_i)$  is locally i.i.d.

The goal of continual learning is to minimize the statistical risk by optimizing the parameters  $\theta$  of the gesture recognition model  $f$  [11]:  $\sum_{t=1}^{\tau} \mathbb{E}_{X^{(t)}, Y^{(t)}} [\mathcal{L}(f_t(X^{(t)}; \theta), Y^{(t)})]$ , with Expectation  $\mathbb{E}$  of the loss function denoted by  $\mathcal{L}$ , the number of tasks seen so far denoted by  $\tau$ ,  $X^{(t)}$  being a set of data samples for task  $t$ ,  $Y^{(t)}$  being the labels correspondingly, and  $f_t$  representing the recognition model for task  $t$ . For the current task  $\tau$ , the statistical risk can be approximated by the empirical risk [11]:  $\frac{1}{N_\tau} \sum_{i=1}^{N_\tau} [\mathcal{L}(f_\tau(x_i^{(\tau)}; \theta), y_i^{(\tau)})]$ .

### 3.3 Large-Scale Data Collection

We collected a large-scale dataset of inertial measurement unit (IMU) data consisting of 6 dimensions (3-axis accelerometer and 3-axis gyroscope).<sup>1</sup>

<sup>1</sup>Figure 1 in the Appendix illustrates exemplary IMU signals collected.

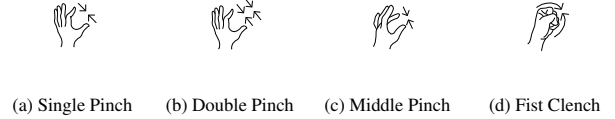


Figure 3: Four dynamic gestures.

1. **Gestures:** We collected data from four dynamic gestures: single pinch, double pinch, middle pinch, and fist clench (see Figure 3). The specific set of gestures was chosen for their distinguishability, and their representation of common hand actions. This selection ensures familiarity, versatility, and intuitive interaction for users. Apple’s AssistiveTouch gestures are also similar to the gesture set.

2. **Participants and Apparatus:** 50 participants were recruited (24 self-identified female, 26 male) with a wide coverage of age range (min = 18, max = 61, mean = 35). The majority of the users were right-handed (N=44). We collected data using a wrist-worn watch-like device equipped with IMU sensors. Initially, we gathered raw data at a high sampling rate of 800 Hz to maximize our dataset’s potential. However, during model training/testing, we found that a sampling rate of 100 Hz was sufficient. Therefore, for the rest of the paper, we exclusively uses 100 Hz data. This rate represents a balanced compromise between a high sampling rate, which would lead to increased power consumption, and a low sampling rate, which could result in the loss of crucial information.

3. **Procedure:** Participants wore a watch-like device with an IMU sensor on their non-dominant hand to collect data. During each session, participants were prompted to perform the target gesture through a chime or vibration, while an experimenter recorded the start and stop times of each gesture. To ensure an accurate representation of real user behavior, randomization was implemented throughout the study. A total of 100 sessions were conducted, with each session including all participants performing 50 instances of a single gesture recurrence for each of the four gestures. These gestures were paired with with 25 contexts, and each of the contexts mimics a real-life context. Examples of the 25 contexts are: *standing/walking with hand up to the chest level/ with hand hanging, riding in a car a passenger, pushing stroller/cart with the other hand, holding a cup or something else in the other hand, sitting at the desk with elbow on the desk and hand in the air/ with arm laying on the desk/ with the elbow on the arm rest and hand in the air/ with the arm on an arm rest, laying down on back/side, lounging on the sofa (horizontal/slouched posture) with hand laying on the sofa or laps, cuddling/arm around someone else, stationary biking, walking up and down stairs, jogging, leaning over, picking something up.*<sup>2</sup> The contexts are also differentiated by if the participant is looking at the device or not. We also collected negative (i.e., non-gesture) examples under various activities such as jogging, running, biking, clapping, driving, waving, cleaning, etc. These examples are similar to the negative examples from [66]. This methodology aimed to capture the natural variability in users’ motions without introducing any repetition bias.

### 4 DESIGN ENGINEERING APPROACH

Conventional machine learning approaches optimize the recognition model to best fit the data once. In an OWGR process, as illustrated in Figure 2, we additionally must optimize the policies by which the recognition model continually updates itself. In the *open-world gesture recognition* process, we outline five stages: (1) device deployment; (2) data logging; (3) data curation; (4) retraining trigger; and (5) updating policy. The final two stages highlight the two crucial design considerations that are integral and critical to an OWGR

<sup>2</sup>Table 1 in the Appendix shows the full list of contexts.

process that allows offline analysis:

1. *When do we update the recognition model?* How do we define the start and stop of a batch, and how frequently should we perform model updates?
2. *How do we update the recognition model?* Which continual learning method provides the ideal trade-off between plasticity and stability, while minimizing processing and memory requirements?

We tackle these two considerations by adapting a design engineering approach [25, 26] to perform a systematic evaluation of an OWGR process without long-term deployment, including user studies over multiple sessions, model deployment and run-time optimization, continuous data logging and collection, and so on. Further, the evaluation of an OWGR process involves system-level parameters beyond the recognition model hyper-parameters.

#### 4.1 Functional Design and Function Carriers of OWGR

Following prior work [25, 26], we begin by identifying the main function (Open-World Gesture Recognition) and its key sub-functions (Data Logging, Data Curation, Retraining Trigger, Updating Policy, Device Deployment), along with their interrelationships.

Having established a function model, we then seek function carriers, or solution principles, and specifically for the Updating Policy function. We evaluate potential function carriers by considering their controllable and uncontrollable parameters through envelope analyses that simulate emergent outcomes as a function of the controllable and uncontrollable parameters in the system. This helps us understand the system requirements for these function carriers to effectively fulfill their functions.

Our study mainly examines two functions, Retraining Trigger and Updating Policy, which both require dedicated online analysis studies. This approach aids in quantitative parameter investigation, thereby enhancing our understanding of continual learning methods for *open-world gesture recognition*. A comprehensive functional design of the entire system is beyond this paper’s scope.

We here study six function carriers. The study of these function carriers will determine which continual learning method to use to update the gesture recognition model. By varying the function carrier, we can explore the design question: *how do we update the recognition model?*. We evaluate the following five different continual learning methods, and a baseline, across the aforementioned families in Section 2.4:

1. **Baseline (finetuning):** For baseline, we use naive finetuning, which optimizes the model trained from previous task on the current task. This method greedily trains each task without considering previous task performance [11].
2. **Learning without Forgetting (LwF):** Learning without Forgetting (LwF) [28] retains knowledge of preceding tasks by means of knowledge distillation.
3. **Synaptic Intelligence (SI):** In SI [70], each synapse accumulates task-relevant information over time and exploits this information to rapidly store new memories without forgetting old ones [70].
4. **PackNet:** PackNet [33] iteratively assigns parameter subsets to consecutive tasks by constituting binary masks. For this purpose, new tasks establish two training phases. The first phase is training the recognition model without altering previous task parameter subsets and then pruning a portion of unimportant free parameters. The second training phase retrains the remaining subset of important subsets. Therefore, PackNet allows explicit allocation of network capacity per task, and therefore inherently supports zero forgetting on previous tasks. However, the disadvantage of PackNet is that the total number of tasks is limited.

5. **Replay:** We use finetuning with an arbitrary replay buffer, which exploits available exemplar memory up to the replay buffer size and incrementally divides equal memory capacity over all previous tasks [11].

6. **Memory Aware Synapses (MAS):** MAS [1] computes the importance of the parameters of a neural network in an unsupervised and online manner. When learning a new task, changes to important parameters can be penalized by the accumulated importance measure for each parameter of the network, effectively preventing important knowledge related to previous tasks from being overwritten [1].

Both SI and MAS are parameter prior-based regularization methods, whereas LwF is a data prior-based regularization method. MAS is a parameter isolation method.

Additionally, we carried out a preliminary offline experiment with internal participants who were asked to perform gestures under two distinct scenarios. In this experiment, we tested the above six different continual learning methods, aiming to adapt the gesture recognition model. The outcomes of this study are presented in Figure 4. This preliminary analysis primarily highlights three key findings:

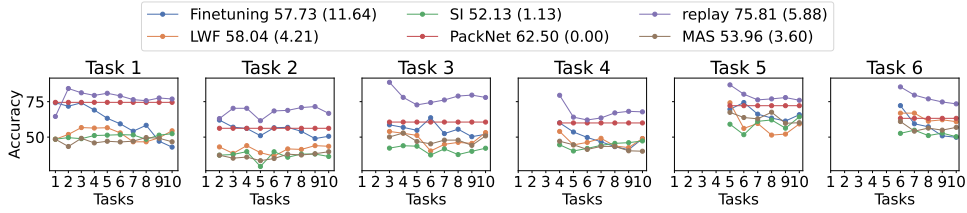
- Naive fine-tuning with new task data leads to significant forgetting of previously learned tasks, underscoring the necessity for continual learning methods to address this issue. This emphasizes the critical nature of the *open-world gesture recognition* paradigm.
- Continual learning methods demonstrate varied effectiveness across different scenarios; a method that shows superior performance in one case may not be the best choice for another. This underlines the importance of conducting comprehensive quantitative analyses on large-scale *open-world gesture recognition* datasets to provide design guidance.
- The execution of this preliminary study required substantial resources and time, as participants needed to perform gestures in a variety of contexts and combine different gestures. This underscores the critical need for online experiments to supplement or replace such extensive offline analyses.

#### 4.2 Surrogate Task Model

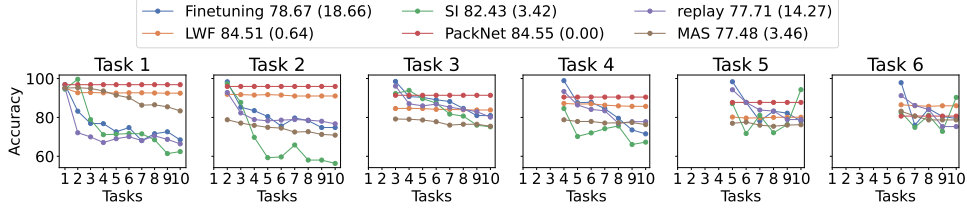
In envelope analysis, a surrogate model is a simplified variant of a complex, computationally involved model. It involves identifying and varying controllable and uncontrollable system parameters to evaluate a system with respect to its emerging outcomes. Controllable parameters can be fine-tuned for optimization, while uncontrollable ones help predict potential performance variations (sensitivity analysis) [25]. Envelope analysis is performed by altering one parameter at a time, with the other parameters held constant.

We here propose a surrogate task model in which we can vary different task settings. The task setting in each case determines the specific definition of each task which is represented by the task descriptor  $t_i$  in a triplet  $(x_i, y_i, t_i)$ . By varying this setting, we can answer the other design question: *when do we update the recognition model?*, because the task setting defines the start and stop of a batch, the size of a batch, and the frequency of model updates, and so on. We have the following parameters in the surrogate task model that determines the task setting:

1. *Granularity of tasks:* The tasks can be either fine-grained or coarse. For example, in the case of *new context*, coarse task granularity is *standing* and fine task granularity is *standing with hand up to the chest level*. This parameter is controllable.
2. *Order of tasks:* By varying the order in which new tasks are presented (e.g. easier-to-harder vs harder-to-easier), we can explore whether the order of tasks affects the learning result. This parameter is uncontrollable because new tasks do not arrive in pre-defined order. The easiness of a task is determined by the preliminary results



(a) *New Context*: Each new task requires one user to perform the same set of gestures under new context, such as standing, walking, riding in a car, laying down, etc. In this context, we have 10 tasks in total, due to size restrictions in presentation, we only show results up to Task 6.



(b) *New User*: Each new task inquires 6 new users to perform the same set of gestures in the same context. This is akin to a model learning to generalize across different users' behavior patterns.

Figure 4: We examine the accuracy of a gesture recognition model, in an *open-world gesture recognition* problem, tested on Task  $n$ , while the model is trained incrementally from Task  $n$  to Task  $N$ .  $N$  is the total number of tasks, and the model is trained by five continual learning methods and one baseline method: Finetuning. For instance, in a subplot with title Task 2, we test the model, progressively trained from Task 2 through Task 10, exclusively on Task 2 and report 9 data points. It lacks the data points from the model trained on Task 1 and tested on Task 2, because the tasks were introduced sequentially; a model trained on Task 1 has not been exposed to Task 2. Consequently, the subplot for Task 2 omits these results. This pattern continues with Task 3 and subsequent tasks, with a subplot of Task  $n$  only reports  $N - n + 1$  data points (per method). The observed deep decline in accuracy from Finetuning in each subplot can be attributed to the model's tendency to forget earlier tasks as it learns new ones, which is referred to catastrophic forgetting. We proposed five continual learning methods to address this catastrophic forgetting problem in *open-world gesture recognition*, namely SI, replay, LWF, PackNet and MAS, which are introduced in detail in Section 4.1. In this Figure, we illustrate the results from performing preliminary experiments on two use cases of *open-world gesture recognition*. We report average accuracy (forgetting) in the legend. Forgetting is the measure of the decrease in accuracy. We discuss these two measures in detail in Section 4.3.

obtained through testing the accuracy of a classification model on that task. A higher accuracy level indicates an easier task.

3. *Number of tasks*: This is equivalent to the number of tasks in total. In the case of *new context*, by investigating the total number of contexts, we can observe the capacity of each continual learning method. This parameter is controllable because we set the maximum total number of tasks to which the system can adapt.

These task settings are applied differently to each of the use cases:

1. *New context*: We incorporate all parameters from the surrogate task model. The default task setting is coarse task granularity provided in a random order. The default number of total tasks is set to 10 (as in [11]).
2. *New user*: We only include the controllable parameter *number of tasks*. Here the number of tasks is the number of total users. The default setting is 15 users, selected at random.

### 4.3 Evaluation Metrics

We use accuracy and forgetting as the evaluation measures for the OWGR model, with definitions adopted from prior work [7] as follows:

1. **Accuracy**: Let  $a_{k,j} \in [0, 1]$  be the gesture recognition accuracy (fraction of correctly classified gesture events) evaluated on the held-out test set of the  $j$ -th task ( $j \leq k$ ) after training the network incrementally from tasks 1 to  $k$ . The accuracy measure at task  $k$  is then defined as  $A_k = \frac{1}{k} \sum_{j=1}^k a_{k,j}$ . The average accuracy measures the plasticity of the system in transferring knowledge from old tasks to new tasks.

2. **Forgetting**: We define forgetting for a particular task as the difference between the maximum knowledge gained about the task throughout the learning process in the past and the knowledge the model currently has about it. This, in turn, gives an estimate of how much the model forgot about the task given its current state (catastrophic forgetting). Following this, we quantify forgetting for the  $j$ -th task after the model has been incrementally trained up to task  $k > j$  as:  $f_j^k = \frac{\mathbb{E} a_{l,j} - a_{k,j}}{a_{l,j}}, \forall j < k$ . Note,  $f_j^k \in [-1, 1]$  is defined for  $j < k$  as we are interested in quantifying forgetting for *previous* tasks. Moreover, by normalizing against the number of tasks seen previously, the average forgetting at  $k$ -th task is written as  $F_k = \frac{1}{k-1} \sum_{j=1}^{k-1} f_{k,j}$ . Lower  $F_k$  implies less forgetting on previous tasks.

In this paper, we do not report accuracy using different metrics, such as F1 scores, because the contribution is not to propose state-of-the-art gesture recognition models. Instead, the focus is on comparing the relative accuracy and the forgetting measures between different continual learning methods to provide design guidance for developing *open-world gesture recognition* methods. Furthermore, given the extensive experiments conducted under various settings in this paper, it is impractical to include and report on different metrics. Therefore, we report on the final accuracy and forgetting, obtained by evaluating each task after learning the entire task sequence.

### 4.4 Implementation Details

We implement the continual learning algorithms based on an open-source generalizing continual learning framework [11] in PyTorch. For data processing, we use a sliding window approach to segment the data for both training data and testing data preparation, we use

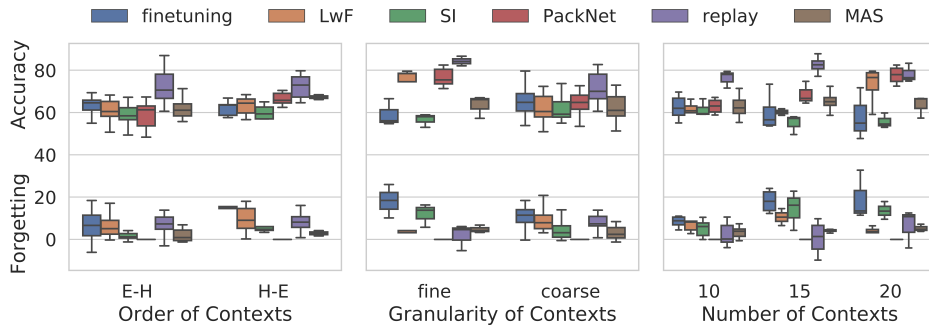


Figure 5: **Effect of Task Setting on Performance for *New Context*** Each sub-figure represents one task setting. Each box plot shows mean, median, and quartile data. a) **Order of Contexts**: Change of context order does not significantly affect average accuracy and forgetting measure for most methods. The X-axis represents a specific order of the tasks. E-H is easy-to-hard, H-E is hard-to-easy. b) **Granularity of Contexts**: Some methods perform particularly better or worse in fine-grained context than coarse context. The X-axis represents that if the context is coarse or fine. c) **Number of Contexts**: Different methods exhibits various performances when number of contexts (tasks) increases. The X-axis represents the total number of contexts (tasks)

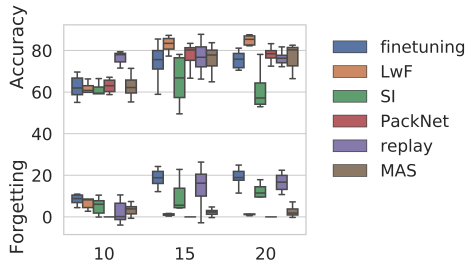


Figure 6: **Effect of Number of Total Tasks for *New User***. A larger number of total users leads to higher accuracy and forgetting measure for most methods. Each box plot shows mean, median, and quartile data. The X-axis represents the total number of users that the model must learn in the *New User* case.

a window size of 120 and a window step size of 60. We adopt the same parameters for debouncing thresholds in [51]. For the hyperparameter search of each continual learning method, we conduct an approach that first tries to decay each hyperparameter separately, and decays all if none of these individual decays to achieve accuracy within the finetuning margin. This process is then repeated until the stability decay criterion is met For the specific setup of the training framework, we define maximal plasticity search with a coarse learning rate grid  $\{1e^{-2}, 5e^{-3}, 1e^{-3}, 5e^{-4}, 1e^{-4}\}$ . We set the finetuning accuracy drop margin to 0.2 and the decaying factor to 0.5. We use a Stochastic Gradient Descent with a momentum of 0.9 and batch size of 128. We use a max of 100 training epochs with early stopping and annealing of the learning rate. That is the learning rate decays with factor 10 after 10 unimproved iterations of the validation accuracy and the training process should terminate after 15 unimproved iterations of training.

We use a simplified version of QuartzNet architecture [24], a state-of-the-art convolutional architecture for speech recognition, for our gesture recognition model described by  $f$  in Equation 1. It is used as the classifier that takes IMU signals as input and predicts among the gesture classes. The selection of the QuartzNet model is based on preliminary experiments. We began with a literature review on state-of-the-art classification models for time series data and tested over 10 models. Our experiments revealed that the QuartzNet model achieves state-of-the-art performance in real-time gesture

recognition, striking an optimal balance between accuracy and latency. We further improved the performance by hyperparameter sweeping making it suitable for IMU data classification. This is attributed to the model’s quantization, which allows it to be compact enough to operate in real-time on smaller devices. To make the model deployable on more memory-constrained devices, we have removed the residual connections, grouped point-wise convolutions, and channel shuffle without scarifying the accuracy performance.<sup>3</sup>

## 5 RESULTS

This section describes the results of varying the surrogate task model for each of the two use cases.

Figure 5 consists of three sub-figures which describe the effect of different task settings separately for the *new context* case. Prior work suggests that an easy-to-hard task ordering might achieve better performance than a hard-to-easy ordering [61]. However, Figure 5 a) shows that the impact of the task order on the average accuracy and forgetting measure is insignificant. We do observe that the variance of the performance scores is considerably smaller for easy-to-hard ordering. In Figure 5 b), when setting the granularity of contexts to fine-grained as the task-setting, we see that finetuning shows a lower accuracy and a higher forgetting measure, whereas methods such as LwF, PackNet, and replay reach a higher accuracy while maintaining a low forgetting measure. A coarse context setting has more training data, as a coarse context consists of multiple fine-grained contexts. This suggests that these continual learning methods can successfully optimize plasticity (transfer knowledge of previous tasks to new tasks) to achieve data-efficiency. Figure 5 c) shows that a lower number of total tasks can stabilize the average performance of the continual learning methods. A larger number of total tasks leads to an increased forgetting measure as the model’s capacity for learning new tasks is limited. However, PackNet and LwF can successfully re-use the previous knowledge to produce an increased accuracy on new tasks. On the other hand, finetuning introduces a significant forgetting measure accompanied by a low accuracy score when the number of tasks is large.

For the *new user* case, we observe from Figure 6 that a larger number of users increases both accuracy and forgetting metrics with finetuning and replay methods. In contrast, most continual learning methods such as PackNet, MAS and LwF achieve high accuracy while maintaining low forgetting measure with larger number of total users. This suggests that, in the *new user* case, PackNet, MAS,

<sup>3</sup>Figure 2 in the Appendix illustrates the details of the model architecture.

and LwF can better transfer knowledge of old users to new users.

## 6 DISCUSSION

### 6.1 Balance between Stability and Plasticity

Continual learning aims to strike a balance between two key objectives: stability and plasticity. Stability refers to the ability to retain previously learned knowledge over time, avoiding catastrophic forgetting. Plasticity, on the other hand, refers to the ability to acquire new knowledge from new data distributions, avoiding overfitting to old data. Stability is typically measured by evaluating the degree of forgetting or performance degradation on previously learned tasks or data distributions. Plasticity is measured by assessing the model's accuracy or performance on new, unseen data distributions. When designing OWGR processes, it is crucial to consider this stability-plasticity trade-off. Design choices and techniques should be evaluated based on their ability to achieve an appropriate balance between these two objectives.

### 6.2 Design Guidelines from Envelope Analysis

#### 6.2.1 When do we update the recognition model?

This question asks for the best task setting for each case. This involves defining the boundaries between tasks and determining when a new task has commenced. Once a new task is detected, a retraining trigger (Figure 2) activates the update of the recognition model using a continual learning method. Thus, we use the results from the envelope analysis from the surrogate task model to answer this question, and we explore which task settings yield the most favorable outcomes. Most of the parameters in the surrogate task model are controllable by the designers. **We advise that a coarse context usually gives better results for the new context case.** A coarse context is also more practical, as it simplifies the activity recognition needed to produce a task descriptor. The accuracy of a task descriptor determines the granularity of tasks; a less accurate descriptor results in a coarser context, while a more precise descriptor leads to a finer-grained context. We previously discussed that the *new context* case may require classification of the current activity context, which can be performed by AR/VR HMDs and wearables via activity detection through different sensors [10, 31]. The precision of state-of-the-art activity classifiers, acting as task descriptors, is sufficiently high that it does not adversely affect the primary gesture recognition model. Therefore, these task descriptors are generalizable descriptors and do not require re-training with new real-world data. **For the new user case, we see the best results when the number of different users is large with LwF and PackNet,** as we observe that a larger number of users increases the overall accuracy for these two methods. **In practice, a model needs to be pre-trained and it is encouraged to pre-train the model on a large-scale dataset if available.**

#### 6.2.2 How do we update the recognition model?

We observe that naive fine-tuning experiences serious catastrophic forgetting and is not suitable for OWGR. For the *new context* case, the replay method outperforms other continual learning methods by a large margin in accuracy. In contrast, PackNet has zero forgetting by its design and still maintains relatively high accuracy. **Overall, for the new context case, the replay method is the best choice to optimize accuracy—memory/privacy constraints permitting—whereas PackNet is a better choice to balance accuracy and forgetting, provided the model capacity is sufficient for the total number of tasks.** For *new user*, the replay method no longer ranked as the top performer in accuracy. Both PackNet and LwF exhibit more stable performance in this case as the difference between the first quartile (Q1) and the third quartile (Q3) is smaller, showing a smaller variance in the performance metrics. **Thus PackNet and LwF is a better choice for the new user case.**

### 6.3 Envelope Analysis: A Practical Alternative to Large-Scale User Studies in OWGR

The primary objective of this research is to introduce a novel method for assessing continual learning methods for OWGR, bypassing the need for large-scale user studies. Researchers often lack resources for such studies, which are more demanding than evaluating conventional gesture recognition models in controlled settings. The proposed approach uses envelope analysis with parameterization of key functions, offering a robust and practical alternative to user studies. By varying parameters, different scenarios can be systematically explored. The extensive and diverse dataset used in this research provides a strong foundation for evaluating the OWGR process, reducing the need for a separate large-scale user study. Although some conclusions drawn from the envelope analysis might appear intuitive, there has been a notable absence of such thorough experiments in the realm of what we would classify as OWGR. Further, assumptions from other, related, areas might not directly translate to gesture recognition problems relevant for mixed reality applications.

### 6.4 OWGR in Mixed Reality

OWGR enables gesture recognition models to adaptively learn new gesture data patterns from diverse contexts and users on-the-fly, eliminating the need for extensive pre-deployment data collection. It facilitates personalization by learning and adapting to individual users' contexts, whether performed in static or dynamic environments. For shared devices among multiple users, such as families, it eliminates recalibration needs by remembering previous users' patterns while continuously learning from new user. This ability to seamlessly learn and retain knowledge across contexts and users ensures a robust, calibration-free experience.

### 6.5 Limitation and Future Work

One limitation in our study is the lower accuracy in gesture recognition. It's worth noting that the accuracy reported in this paper represents the final model's accuracy on the last task, following the entire task sequence (see Section 4.3). The accuracy diminished from the initial task to the final task due to catastrophic forgetting. This further encourages us to refine our continual learning algorithms as future work. Another contributing factor is that our training algorithm wasn't explicitly tailored for optimal accuracy. Techniques such as data augmentation, feature engineering, and model pre-training, which are commonly used to enhance model performance [49, 51, 66], were not employed in our study. This decision was intentional, as our primary aim was to isolate the effects of the envelope analysis. By excluding these optimization techniques, we ensured that any observed effects on the final results were solely due to changes in the parameters of the envelope analysis.

## 7 CONCLUSION

This paper presents a design engineering approach, using a large-scale dataset in combination with envelope analysis, to explore continual learning methods for the *open-world gesture recognition* process. We discuss the significant catastrophic forgetting observed in the baseline fine-tuning approach and examine the positive and negative qualities of various continual learning methods. To assist other researchers and developers we present guidelines derived from the extensive data collection and the computational experiments. Intended for developers and designers of wrist-worn gesture-sensing systems, our guidelines provide an alternative to traditional large-scale user studies, lowering the barriers and promoting a more refined OWGR process. This paper is also a demonstration of how a design engineering methodology, in particular envelope analysis, offers a new route for system evaluations in this area, potentially inspiring other researchers in mixed reality to adopt similar methods for real-world assessment challenges.



## REFERENCES

- [1] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.
- [2] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller. State of the art of virtual reality technology. In *2016 IEEE aerospace conference*, pp. 1–19. IEEE, 2016.
- [3] A. Bendale and T. Boulton. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1893–1902, 2015.
- [4] G. Buckingham. Hand tracking for immersive virtual reality: opportunities and challenges. *Frontiers in Virtual Reality*, 2:728461, 2021.
- [5] P. Buzzega, M. Boschini, A. Porrello, and S. Calderara. Rethinking experience replay: a bag of tricks for continual learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2180–2187. IEEE, 2021.
- [6] B. Caramiaux, M. Donnarumma, and A. Tanaka. Understanding Gesture Expressivity through Muscle Sensing. *ACM Transactions on Computer-Human Interaction*, 21(6):1–26, Jan. 2015. doi: 10.1145/2687922
- [7] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.
- [8] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato. Continual learning with tiny episodic memories. 2019.
- [9] K.-Y. Chen, S. N. Patel, and S. Keller. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 1504–1514. ACM, San Jose California USA, May 2016. doi: 10.1145/2858036.2858125
- [10] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar. A survey on activity detection and classification using wearable sensors. *IEEE Sensors Journal*, 17:386–403, 2017.
- [11] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [12] A. Dementyev and J. A. Paradiso. WristFlex: low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pp. 161–166. ACM, Honolulu Hawaii USA, Oct. 2014. doi: 10.1145/2642918.2647396
- [13] S. K. Feiner. Augmented reality: A new way of seeing. *Scientific American*, 286(4):48–55, 2002.
- [14] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.
- [15] R. M. French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.
- [16] M. Georgi, C. Amma, and T. Schultz. Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing. In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*, pp. 99–108. SCITEPRESS - Science and Technology Publications, Lisbon, Portugal, 2015. doi: 10.5220/0005276900990108
- [17] N. Gillian and J. A. Paradiso. The Gesture Recognition Toolkit. In S. Escalera, I. Guyon, and V. Athitsos, eds., *Gesture Recognition*, pp. 497–502. Springer International Publishing, Cham, 2017. Series Title: The Springer Series on Challenges in Machine Learning. doi: 10.1007/978-3-319-57021-1\_17
- [18] F. Hu, P. He, S. Xu, Y. Li, and C. Zhang. FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–24, June 2020. doi: 10.1145/3397306
- [19] Y. Iravanchi, M. Goel, and C. Harrison. BeamBand: Hand Gesture Sensing with Ultrasonic Beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–10. ACM, Glasgow Scotland UK, May 2019. doi: 10.1145/3290605.3300245
- [20] D. Isele and A. Cosgun. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [21] S. Jiang, B. Lv, W. Guo, C. Zhang, H. Wang, X. Sheng, and P. B. Shull. Feasibility of wrist-worn, real-time hand, and surface gesture recognition via semg and imu sensing. *IEEE Transactions on Industrial Informatics*, 14(8):3376–3385, 2017.
- [22] P.-G. Jung, G. Lim, S. Kim, and K. Kong. A Wearable Gesture Recognition Device for Detecting Muscular Activities Based on Air-Pressure Sensors. *IEEE Transactions on Industrial Informatics*, 11(2):485–494, Apr. 2015. Conference Name: IEEE Transactions on Industrial Informatics. doi: 10.1109/TII.2015.2405413
- [23] M. Kim, J. Cho, S. Lee, and Y. Jung. IMU Sensor-Based Hand Gesture Recognition for Human-Machine Interfaces. *Sensors*, 19(18):3827, Sept. 2019. doi: 10.3390/s19183827
- [24] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6124–6128. IEEE, 2020.
- [25] P. O. Kristensson, J. Lilley, R. Black, and A. Waller. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. *CHI '20*, p. 1–11. Association for Computing Machinery, New York, NY, USA, 2020.
- [26] P. O. Kristensson and T. Müllners. Design and analysis of intelligent text entry systems with function structure models and envelope analysis. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2021.
- [27] G. Laput and C. Harrison. Sensing Fine-Grained Hand Activity with Smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13. ACM, Glasgow Scotland UK, May 2019. doi: 10.1145/3290605.3300568
- [28] Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.
- [29] J. Lien, N. Gillian, M. E. Karagozler, P. Amihhood, C. Schwesig, E. Olsson, H. Raja, and I. Poupyrev. Soli: ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics*, 35(4):1–19, July 2016. doi: 10.1145/2897824.2925953
- [30] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, Dec. 2009. doi: 10.1016/j.pmcj.2009.07.007
- [31] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1942–1950, 2016.
- [32] A. Mallya, D. Davis, and S. Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 67–82, 2018.
- [33] A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.
- [34] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *ArXiv*, abs/0902.3430, 2009.
- [35] J. McIntosh, A. Marzo, and M. Fraser. SensIR: Detecting Hand Gestures with a Wearable Bracelet using Infrared Transmission and Reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 593–597. ACM, Québec City QC Canada, Oct. 2017. doi: 10.1145/3126594.3126604
- [36] J. McIntosh, C. McNeill, M. Fraser, F. Kerber, M. Löchtfeld, and A. Krüger. Empress: Practical hand gesture classification with wrist-mounted emg and pressure sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 2332–2342, 2016.

- [37] S. J. McKenna and K. Morrison. A comparison of skin history and trajectory-based representation schemes for the recognition of user-specified gestures. *Pattern Recognition*, 37(5):999–1009, May 2004. doi: 10.1016/j.patcog.2003.09.007
- [38] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 1515–1525. ACM, San Jose California USA, May 2016. doi: 10.1145/2858036.2858580
- [39] F. S. Parizi, E. Whitmire, and S. Patel. AuraRing: Precise Electromagnetic Finger Tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–28, Dec. 2019. doi: 10.1145/3369831
- [40] A. Pritzel, B. Uria, S. Srinivasan, A. P. Badia, O. Vinyals, D. Hassabis, D. Wierstra, and C. Blundell. Neural episodic control. In *International Conference on Machine Learning*, pp. 2827–2836. PMLR, 2017.
- [41] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.
- [42] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll. Static and dynamic hand-gesture recognition for augmented reality applications. In *International Conference on Human-Computer Interaction*, pp. 728–737. Springer, 2007.
- [43] H. Ren, A. Sootla, T. Jafferjee, J. Shen, J. Wang, and H. Bou-Ammar. Reinforcement learning in presence of discrete markovian context evolution. *arXiv preprint arXiv:2202.06557*, 2022.
- [44] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.
- [45] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- [46] K. M. Sagayam and D. J. Hemanth. Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality*, 21(2):91–107, 2017.
- [47] T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*, p. 167. ACM Press, Victoria, BC, Canada, 2009. doi: 10.1145/1622176.1622208
- [48] S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.
- [49] J. Shen, J. Dudley, and P. O. Kristensson. The imaginative generative adversarial network: Automatic data augmentation for dynamic skeleton-based hand gesture and human action recognition. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, p. 1–8. IEEE Press, 2021.
- [50] J. Shen, J. Dudley, and P. O. Kristensson. Simulating realistic human motion trajectories of mid-air gesture typing. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 393–402, 2021. doi: 10.1109/ISMAR52148.2021.00056
- [51] J. Shen, J. J. Dudley, M. George, and P. O. Kristensson. Gesture spotter: A rapid prototyping tool for key gesture spotting in virtual and augmented reality applications. In *IEEE transactions on visualization and computer graphics*, 2022.
- [52] J. Shen, B. Yang, J. J. Dudley, and P. O. Kristensson. Kwickchat: A multi-turn dialogue system for aac using context-aware sentence generation by bag-of-keywords. In *27th International Conference on Intelligent User Interfaces, IUI '22*, p. 853–867. Association for Computing Machinery, 2022.
- [53] Y. SHI, Y. LI, X. FU, M. Kaibin, and M. Qiguang. Review of dynamic gesture recognition. *Virtual Reality & Intelligent Hardware*, 3(3):183–206, 2021.
- [54] N. Siddiqui and R. H. Chan. Multimodal hand gesture recognition using single imu and acoustic measurements at wrist. *Plos one*, 15(1):e0227039, 2020.
- [55] P. Strohmeier, R. Vertegaal, and A. Girouard. With a flick of the wrist: stretch sensors as lightweight input for mobile devices. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pp. 307–308. ACM, Kingston Ontario Canada, Feb. 2012. doi: 10.1145/2148131.2148195
- [56] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2017.
- [57] H. Truong, S. Zhang, U. Muncuk, P. Nguyen, N. Bui, A. Nguyen, Q. Lv, K. Chowdhury, T. Dinh, and T. Vu. CapBand: Battery-free Successive Capacitance Sensing Wristband for Hand Gesture Recognition. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pp. 54–67. ACM, Shenzhen China, Nov. 2018. doi: 10.1145/3274783.3274854
- [58] J. Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.
- [59] V. N. Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.
- [60] M. Vrigkas, C. Nikou, and I. A. Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.
- [61] X. Wang, Y. Chen, and W. Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [62] Z. Wang, Q. She, T. Chalasani, and A. Smolic. Catnet: Class incremental 3d convnets for lifelong egocentric gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 230–231, 2020.
- [63] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.
- [64] E. Wu, Y. Yuan, H.-S. Yeo, A. Quigley, H. Koike, and K. M. Kitani. Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-worn Camera via Dorsum Deformation Network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pp. 1147–1160. ACM, Virtual Event USA, Oct. 2020. doi: 10.1145/3379337.3415897
- [65] X. Xu, A. Dancu, P. Maes, and S. Nanayakkara. Hand range interface: information always at hand with a body-centric mid-air input surface. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 1–12. ACM, Barcelona Spain, Sept. 2018. doi: 10.1145/3229434.3229449
- [66] X. Xu, J. Gong, C. Brum, L. Liang, B. Suh, K. Gupta, Y. Agarwal, L. Lindsey, R. Kang, B. Shahsavari, T. Nguyen, H. Nieto, S. E. Hudson, C. Maalouf, S. Mousavi, and G. Laput. Enabling hand gesture customization on wrist-worn devices. 2022.
- [67] L. Yang, J. Huang, T. Feng, W. Hong-An, and D. Guo-Zhong. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019.
- [68] H.-S. Yeo, E. Wu, J. Lee, A. Quigley, and H. Koike. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 963–971. ACM, New Orleans LA USA, Oct. 2019. doi: 10.1145/3332165.3347867
- [69] J. Yoon, E. Yang, J. Lee, and S. J. Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.
- [70] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.
- [71] Y. Zhang and C. Harrison. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 167–173. ACM, Charlotte NC USA, Nov. 2015. doi: 10.1145/2807442.2807480
- [72] Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [73] Z. Zhang, Z. Tian, Z. Mu, and Y. Liu. Application of fmcw radar for dynamic continuous hand gesture recognition. 2018.